Getting Started With Matlab
Last updated August 24, 2006

If you encounter any spelling or grammatical errors, please let me know, I am writing this up on the fly!

1 Introduction

Getting started is always the most difficult part of learning any new software package. Hopefully this will help clarify things a bit. Using this along with the Matlab Programming .pdf will really improve your learning curve.

It will be assumed that you have already installed Matlab on your computer.

2 Loading Matlab

If you are running Windows or a similar OS, there should be a Matlab icon on your desktop. Double clicking on it will start the program.

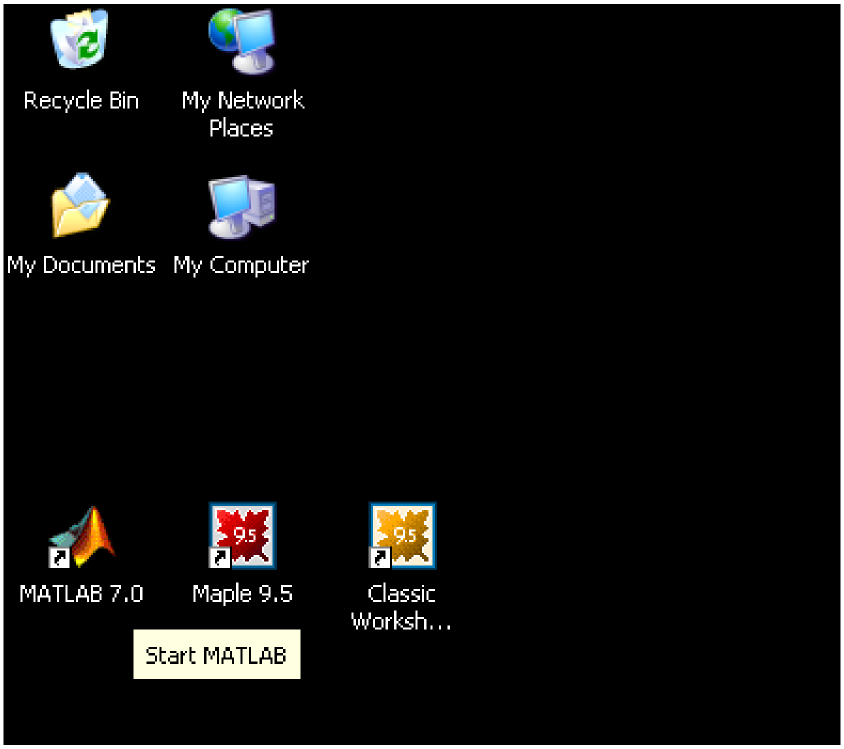
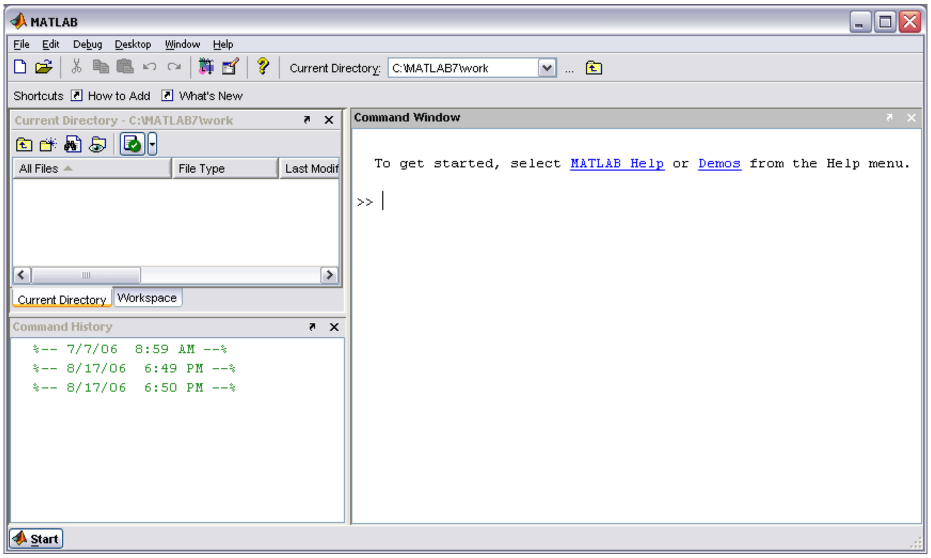


Figure 1: The Matlab Icon on a Windows XP desktop

When Matlab first loads up, it should look something like this:



Current Directory: C:\MATLAB7\work

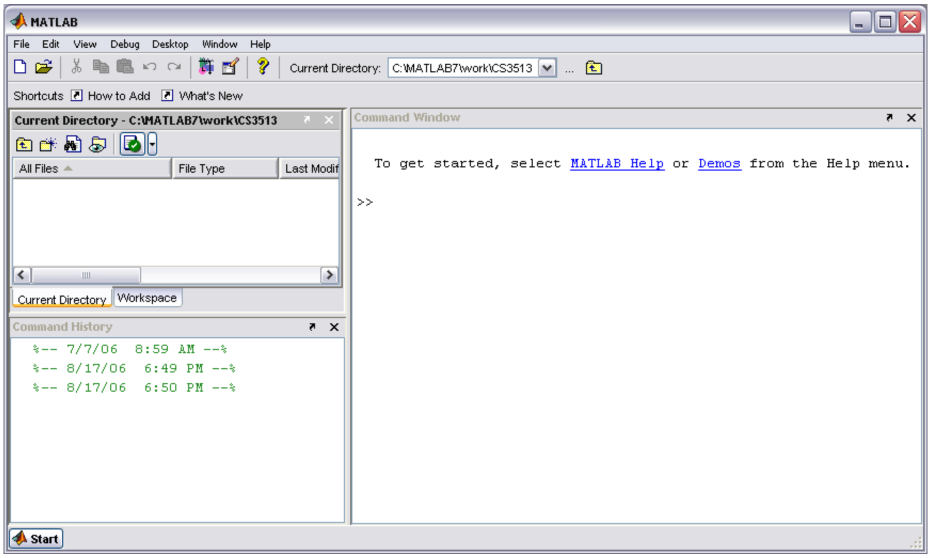
```
To get started, select MATLAB Help or Demos from the Help menu.

>>
```

```
%-- 7/7/06  8:59 AM --%
%-- 8/17/06  6:49 PM --%
%-- 8/17/06  6:50 PM --%
```

Figure 2: The Matlab program window

The next thing we will want to do is create a new folder in which to store our work. We will call the folder CS3513. To do this, we must right clink in the upper left window which is labelled '*Current Directory*' in the white space and select the option *New* → *Folder*. There will now be a new folder in the *Current Directory* window, name it whatever you want, I chose to name it CS3513. Double click on this directory like you would in Windows Explorer. Once you have done this, notice the path on the status bar of the *Current Directory* window is now pointed to the new directory.



Current Directory: C:\MATLAB7\work\CS3513

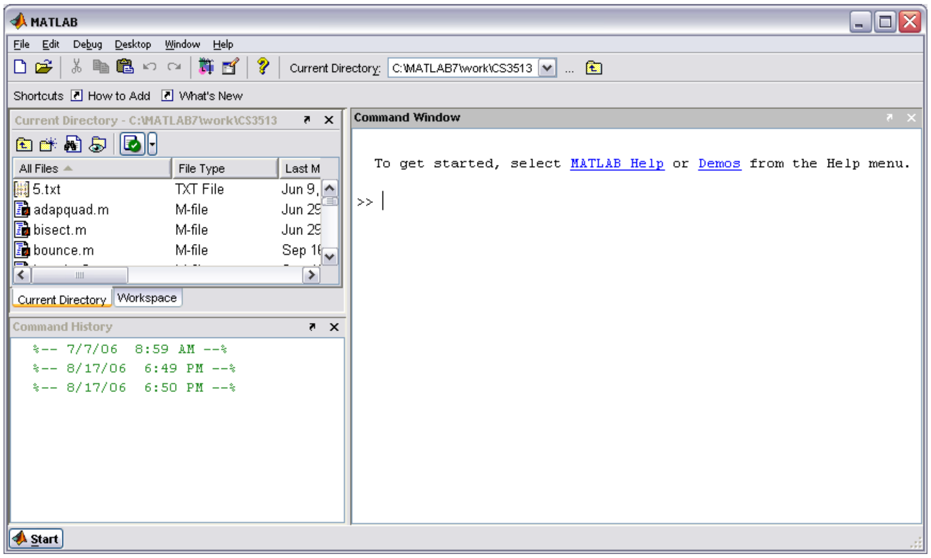
```
To get started, select MATLAB Help or Demos from the Help menu.

>>
```

```
%-- 7/7/06  8:59 AM --%
%-- 8/17/06  6:49 PM --%
%-- 8/17/06  6:50 PM --%
```

Figure 3: The new CS3513 folder

Now we are going to do a test run. So we are going to take the files from our CD that came with the *Numerical Analysis* by Timothy Sauer book and copy those files to the working directory. The path to which you are going to copy the files from the CD is going to be something like C:\ Matlab7 \ work \ CS3513. Note that this is for Windows on a standard install. Your path may vary. Once copied, you will notice that the files show up in the *Current Directory* window.


MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work\CS3513
Shortcuts How to Add What's New
Current Directory - C:\MATLAB7\work\CS3513
Command Window
Current Directory Workspace
Command History
Start

All Files	File Type	Last M
5.txt	TXT File	Jun 9,
adapquad.m	M-file	Jun 29
bisect.m	M-file	Jun 29
bounce.m	M-file	Sep 18

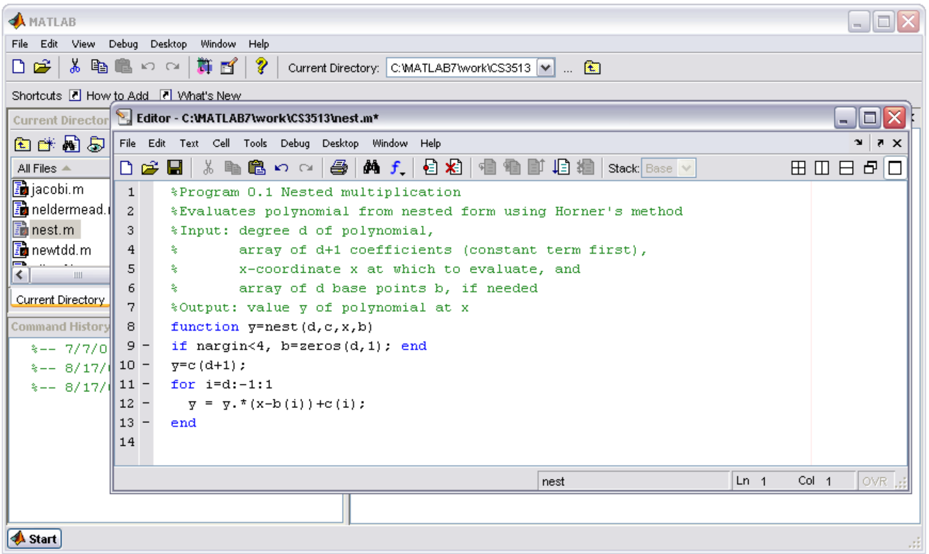
```
%-- 7/7/06  8:59 AM --%
%-- 8/17/06  6:49 PM --%
%-- 8/17/06  6:50 PM --%
```

```
  To get started, select MATLAB Help or Demos from the Help menu.

>> 
```

Figure 4: All the files are in the *Current Directory* window

The M-file file type denotes a Matlab Program file. You may run M-files from the command prompt if you like. You may also double click on the M-files so that they will pop up in a new editor window. As an example, we will look at the code for the nest program encountered in section 0.1 (see page 3 of the text). To do this, double click on the nest.m file and it will pop up in a new editor window. Remember, always click on the .m files, not the .asv files, those are saved backup copies.


MATLAB
File Edit View Debug Desktop Window Help
Current Directory: C:\MATLAB7\work\CS3513
Shortcuts How to Add What's New
Editor - C:\MATLAB7\work\CS3513\nest.m*
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
All Files
jacobi.m
neldermead.m
nest.m
newtdd.m
Current Directory
Command History
nest
Ln 1
Col 1
OVR
Start

```
%Program 0.1 Nested multiplication
%Evaluates polynomial from nested form using Horner's method
%Input: degree d of polynomial,
%       array of d+1 coefficients (constant term first),
%       x-coordinate x at which to evaluate, and
%       array of d base points b, if needed
%Output: value y of polynomial at x
function y=nest(d,c,x,b)
if nargin<4, b=zeros(d,1); end
y=c(d+1);
for i=d:-1:1
  y = y.*(x-b(i))+c(i);
end
```

Figure 5: The M-file nest.m loaded in an Editor Window

Once the editor window has opened up, you can view the code for the nest.m file. Comments are always started with at percent sign, and appear green. It is always a good idea to read the comments at the beginning of a program, it will help you figure out how to call the program. The first line of executable code is actually on line 8. It reads as follows:

```
function y=nest(d,c,x,b)
```

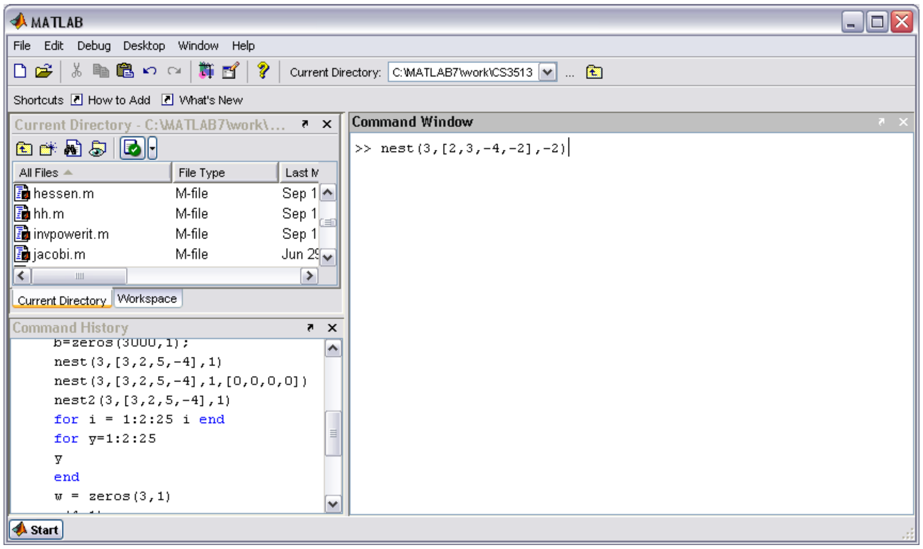
This implies that to call this function, you will have to type into the command window of Matlab a command of the form:

```
nest(d,c,x,b)
```

where d, c x and b are predefined variables. According to the comments, d is the degree of the polynomial, c is an array of polynomial coefficients (notice there must be d+1 entries in the array), x is the point at which to evaluate, and b is the base point. If the base point array b is not used, it will be assumed to be identically zero.

Consider the polynomial $p(x) = 2 + 3x - 4x^2 - 2x^3$, to be evaluated at $x = -2$. To run the nest program, simply type the following command into the Command Window:

$$\text{nest}(3, [2, 3, -4, -2], -2)$$


MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work\CS3513
Shortcuts How to Add What's New
Current Directory - C:\MATLAB7\work\...
Current Directory
Workspace
Command History
Command Window
Start

All Files	File Type	Last M
hessen.m	M-file	Sep 1
hh.m	M-file	Sep 1
invpowerit.m	M-file	Sep 1
jacobi.m	M-file	Jun 29

```
b=zeros(3000,1);
nest(3,[3,2,5,-4],1)
nest(3,[3,2,5,-4],1,[0,0,0,0])
nest2(3,[3,2,5,-4],1)
for i = 1:2:25 i end
for y=1:2:25
y
end
w = zeros(3,1)
```

```
>> nest(3,[2,3,-4,-2],-2)
```

Figure 6: Calling the nest.m file through the Command Window

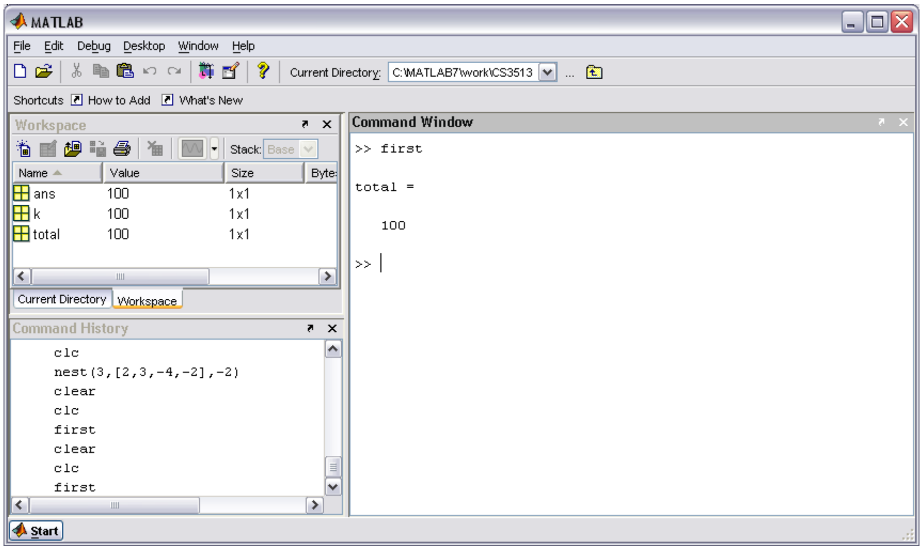
Be sure not to place a semicolon after the command, so that the answer will be displayed. Upon pressing enter, Matlab should display the output as:

```
ans =
     -4
```

Next we will create our own Matlab M-file. Close the current Editor Window and in the main Matlab window, select *File* → *New* → *M-File*. The first thing to do with your new M-File is save it. Call it whatever you want, I will name it *first.m*. We are going to perform a simple *for* loop, so we will not even need to define a function. Enter the following code, or download the *first.m* file from my webpage.

```
total = 0;
for k = 1:100
    total = total + 1;
end
total
```

To run this program, simply type *first* into the Command Window. Notice the answer is the number of times the loop is executed.



Current Directory: C:\MATLAB7\work\CS3513

Name	Value	Size	Byte
ans	100	1x1	
k	100	1x1	
total	100	1x1	

```
    clc
    nest(3,[2,3,-4,-2],-2)
    clear
    clc
    first
    clear
    clc
    first
```

```
>> first

total =

   100

>> |
```

Figure 7: Calling and executing the first.m file through the Command Window

In the workspace window on the main Matlab console, change views to the workspace view and notice the variables that are now visible.